

ネットワーク管理者に情報が漏洩しない セキュアグループ通信方式の提案

163430015 棚田 慎也

渡邊研究室

1. はじめに

ネットワーク技術の発展により、インターネットを介して情報を共有する機会が増加している。中でもグループ内での情報共有のためにグループ通信におけるセキュリティが極めて重要な項目として挙げられる。そのセキュリティを満たすため、グループ鍵を用いて相手認証やコンテンツを暗号化する方法が一般的に用いられる。このとき鍵管理要件と呼ばれる条件を満たすことが必須とされている。しかし、メッセージアプリケーションのセキュリティ評価を行っている非営利団体 EFF(Electric Frontier Foundation)¹によると現状の主流なメッセージアプリケーションのセキュリティが極めて脆弱であることが指摘されている。EFFの評価項目の中には、鍵管理要件以外にグループ鍵が管理者が読めないように暗号化されているかどうかがあり、多くのシステムではこの項目を満たしていない。そこで本稿では、2種類の乱数からグループ鍵を生成することにより鍵管理要件とEFFの評価項目の両者を満たしたグループ通信方式を提案する。

2. 鍵管理要件と既存技術

2.1 鍵管理要件

グループ鍵管理プロトコルは機密性や認証に必要なデータを最新の暗号化状態でグループメンバに送信することが重要であり、一般的に以下のような鍵管理要件が存在する。

- 鍵はあらかじめ定めた期間で定期的に更新を行う。
- 鍵データは厳重に保管され、正規ソースからのみ入手可能であり、正しいグループメンバのみに送られる。
- 鍵管理プロトコルはリプレイ攻撃やDoS(Denial of Service)攻撃に対して安全である。
- 参加や退会が容易に可能であり、新たに参加したメンバがグループに参加する前の鍵データにアクセスできない(後方秘匿性)。また退会したメンバがそれ以降の鍵データにアクセスできない(前方秘匿性)。

2.2 既存技術 GSAKMP

グループ鍵を用いたグループ通信におけるセキュリティフレームワークとしてGSAKMP(Group Secure Association Key Management Protocol[1]:RFC4535)が標準化されている。GSAKMPではグループオーナーGO、鍵サーバGCKS、グループメンバGMの3つの主要な要素でグループ管理を分散している。GOによって作成されたセキュリティポリシーを基にアクセス制御を実行する。そのセキュリティポリシーに基づきGCKSはGMの管理やグループ鍵の生成を行う。GSAKMPではGCKSだけでなくGMも公開鍵証明書を所有しGCKSと各GM間の相互認証を確実に行う。GSAKMPにおけるグループ鍵共有シーケンスを図1に示す。新たに参加するユーザはGOから招待されていることが前提である。招待されたユーザはGCKSへRequest to Joinを送信する。GCKSはユーザの公開鍵

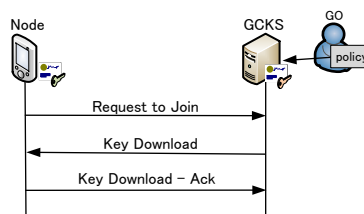


図1: GSAKMPにおけるグループ鍵共有シーケンス

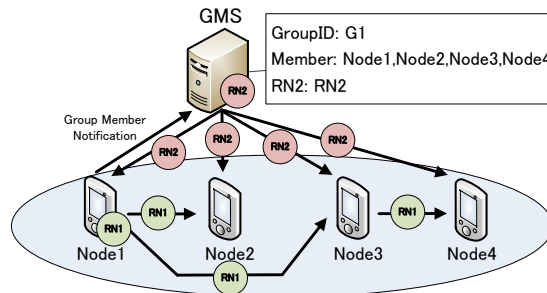


図2: 提案方式のシステム構成

証明書を確認しグループ鍵を配布する。ユーザは応答を返すことによってこのタイミングで当該グループのメンバとなる。

しかし、この方式はGCKSがグループ鍵を配布しているため、グループ鍵管理者に通信内容を読み取られる恐れがある。また、GOが悪意のあるメンバをグループに混入させることも可能である。

3. 提案方式

3.1 提案方式の原理とシステム構成

本提案はサーバ管理者にも情報が漏洩しないセキュアなグループ通信方式を実現することが目的である。この目的を達成するために、生成元が異なる2種類の乱数RN1,RN2を異なる配送経路でグループメンバへ配送し、その2種類の乱数から新たなグループ鍵GKを生成する。鍵管理サーバGMS(Group Management Server)の管理者はGKを知ることができないため、EFFの評価項目を満たすことができる。

図2に提案方式のシステム構成を示す。提案方式はGMSとユーザ端末から成る。GMSはグループ情報のデータベース管理、乱数RN2の生成管理および配布を行い、あらかじめ設定してあるタイミングでRN2の更新を行う。ユーザ端末は乱数RN1の生成、共有およびGKの生成を行う。また、グループメンバの招待はユーザ端末が実行し、GMSの管理者はグループメンバの管理には関与しない。なお公開鍵

¹<https://www.eff.org/>

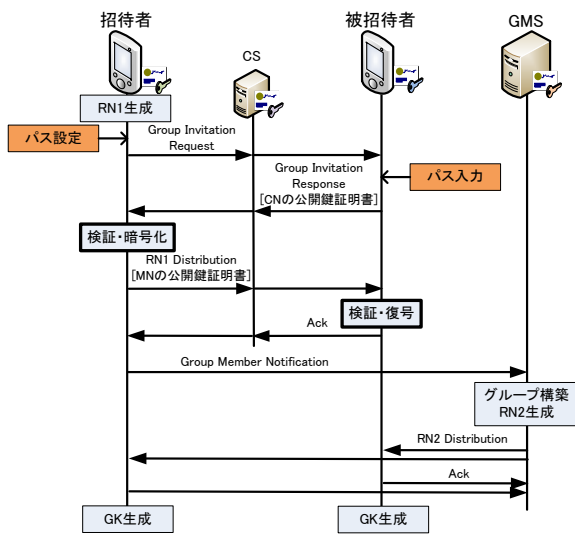


図 3: 公開鍵証明書を用いた鍵共有シーケンス

はRSA(鍵長 1024 ビット以上), ハッシュ関数はSHA256を使用し, 暗号化アルゴリズム上はセキュリティに課題がないことを前提とする.

3.2 鍵共有方式

提案方式の実現方法として, 公開鍵証明書をユーザ端末に所有させる方法と, エンドツーエンド通信が可能なセキュアネットワークを使用する方法の2通りが考えられる. 公開鍵証明書を用いる場合, 乱数配送に一般的なコミュニケーションサーバを使用することができるが, 公開鍵証明書の取得や管理にコストがかかる. 一方, エンドツーエンド通信が可能なネットワークの場合, ユーザ端末が公開鍵証明書を取得する必要はないが, エンドツーエンド通信が可能なセキュアネットワークを準備する必要がある. このようなネットワークの例として NTMobile[2] がある.

図 3 に公開鍵証明書を用いた鍵共有シーケンスを示す. 招待を行うユーザを招待者, 招待されたユーザを被招待者とする. RN1 を公開鍵で暗号化できるため, ユーザ間の通信には一般のコミュニケーションサーバ CS を経由した方法であっても構わない. 図 3 において, 最初の招待者の端末は RN1 を生成する. 招待者と被招待者はあらかじめパスワードを共有しておくことが望ましい. 被招待者は招待者から権限を与えられた場合, 新たに招待者となることができる. 招待者は被招待者へグループ ID とともに Group Invitation Request を送信する. この時点では DoS 攻撃の対象となることを防ぐため, 招待者側の公開鍵証明書は送付しない. 被招待者は事前共有したパスワードを入力し, 被招待者の公開鍵証明書とともに Group Invitation Response として応答を返す. 招待者は被招待者の公開鍵証明書の検証を行い, 正しい場合 RN1 を被招待者の公開鍵で暗号化する. そして招待者の公開鍵証明書とともに暗号化した RN1 を RN1 Distribution として被招待者へ送信する. 被招待者は招待者の公開鍵証明書の検証を行い, 正しい場合 RN1 を自身の秘密鍵で復号し, Ack を返す. 以上により RN1 の共有が完了する. RN1 は新たなメンバが加わった場合も半永久的に引き継がれる.

次に招待者は GMS へ Group Member Notification を送信し, グループ情報の変更を通知する. GMS はグループ管理用のテーブルの更新を行う. また当該グループの

表 1: 提案方式の証明書検証と暗号化/復号の計測結果

	招待者側 [ms]	被招待者側 [ms]
証明書検証	67.24	67.83
暗号化/復号	91.02	48.82

表 2: 関連技術と提案方式の比較

	項目 1	項目 2	項目 3
GSAKMP	×	○	○
ChatSecure	○	○	×
提案方式	○	○	○

RN2 を作成し, グループメンバへそれぞれ配布する (RN2 Distribution). RN2 はあらかじめ設定された更新期間を経過した場合, もしくはグループメンバが新たに参加や退会をした場合にその都度更新を行いグループメンバへ配布される.

RN1, RN2 を取得した各ユーザはハッシュ関数を用いて [RN1|RN2|GroupName] のハッシュ値を取り, グループ共通鍵 GK を生成する. これにより同一グループメンバのみが GK を所有し, この内容は管理者にもわからない.

4. 実装と評価

4.1 実装と性能評価

提案方式の一部を実装し, 処理時間の計測を行った. 仮想環境において図 3 に相当するシステムを構築し, GMS とユーザ端末に公開鍵証明書を配布した. 測定箇所は図 3 の検証・暗号化と検証・復号の4箇所であり, 各試行回数 10 回の平均時間を取得した. その結果を表 1 に示す. 証明書検証と公開鍵による暗号化/復号に多くの時間を要するが招待時のみであり, 実用上問題がないと言える.

4.2 関連技術との比較

表 2 に関連技術との比較を示す. 項目は以下の3つである. 比較対象は GSAKMP, ChatSecure である. ここで ChatSecure は EFF の評価項目を全て満たすメッセージアプリケーションであるが 1 対 1 通信しか行えない.

1. 管理者が読めないように暗号化されているか.
2. 鍵管理要件を満たしているか.
3. グループ通信が可能であるか.

GSAKMP は鍵管理要件を満たしているが, サーバがグループ鍵配布を行っているため, 項目 1 を満たしていない. ChatSecure は 1 対 1 通信のみであり項目 3 を満たせない. 提案方式では全項目を満たしており, 最も安全である.

5. まとめ

2 種類の乱数からグループ鍵を生成するセキュアグループ通信方式を提案し, サーバ管理者にも情報が漏洩しないグループ通信が可能であることを示した. 提案方式の一部を実装し, 動作検証と計測の結果, 実用上問題がない時間で実行できることを確認した.

参考文献

- [1] H. Harne, U. Meth, A. Colegrove, G. Gross: GSAKMP: Group Secure Association Key Management Protocol, RFC4535, IETF (2006).
- [2] 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊晃: IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価情報処理学会論文誌, Vol.54, No.10, pp.2288-2299 (2013).